Available online at www.sciencedirect.com

SCIENCE DIRECT®

Nuclear Instruments and Methods in Physics Research B 215 (2004) 577–580

NIM B Beam Interactions with Materials & Atoms

www.elsevier.com/locate/nimb

A remote data acquisition and control system for Mössbauer spectroscopy

Qingguo Zhou [a,b], Li Wang [b,c,*], Yanlong Wang [d], Hong Zhao [b], Rongjie Zhou [b,c]

[a] *Network Center, Lanzhou University, Lanzhou, Gansu 730000, PR China*

[b] *Department of Physics, Lanzhou University, Tianshui Road 298, Lanzhou, Gansu 730000, PR China*

[c] *Key Laboratory for Magnetism and Magnetic Material of Ministry of Education, Lanzhou University, Lanzhou, Gansu 730000, PR China*

[d] *Department of Electronic Engineering, Tsinghua University, Beijing 100084, PR China*

Received 7 November 2002; received in revised form 20 August 2003

Abstract

A remote data acquisition and control system for Mössbauer spectroscopy based on an embedded computer with the Mini Real-Time Linux operating system is presented. This system can be accessed by an Internet browser or a Java application program, which is designed especially for this purpose. So controlling this system is simple and the interface is user friendly. The components of this system can easily be obtained. So it could be built in most laboratories. We have succeeded in designing and developing this system, as well as using the system at the Key Laboratory for Magnetism and Magnetic Material of Ministry of Education, Lanzhou University, PR China.

PACS: 76.80.+g; 07.05.Hd

Keywords: Mössbauer spectroscopy; Data acquisition

1. Introduction

Mössbauer spectroscopy (MS) has become a popular tool in many research fields. In the 1970s, a standard Mössbauer spectrometer [1] made use of the memory of a multi-channel analyzer (MCA) in the multi-scalar mode. A multi-channel analyzer (MCA) is expensive and difficult to operate, particularly if remote access is required. Later, there have been moves towards replacing the MCA by microprocessors [2]. More recently, some systems based on a PC were built [3,6]. However, neither of these systems do support remote accessing and controlling, which is a desire with modern collaborating research demands. The need for remote access to experiment data is a consequence of the current trend in nuclear physics experiments. It is worth noting that collaborating researchers could access and control the data acquisition system simply by remotely connecting to the experimental facility and using the local computer system, but some assistance will be needed. Furthermore, the

* Corresponding author. Address: Department of Physics, Lanzhou University, Tianshui Road 298, Lanzhou, Gansu 730000, PR China. Tel.: +86-0931-8912025/0185; fax: +86-0931-8912688.

E-mail address: wangl@lzu.edu.cn (L. Wang).

0168-583X/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.nimb.2003.08.043

data acquisition system for Mössbauer spectroscopy is a "strong" real-time system. The recent development of Real-Time Linux [7] as a reliable operating system for embedded applications, implementing multi-task and multi-user capabilities, has provided a good environment to implement such systems. In this paper, we have designed a system of which the hardware components are easy to obtain. This system has been developed for an embedded computer based on X86 architecture, and has a high running speed. It is very reliable, inexpensive and economical, and has never been reported before.

2. Description of the system

In these Mössbauer spectroscopy (MS) experiments, the source is driven by a triangular or sinusoidal wave form with high frequency (60–100 Hz). The spectrum is a record of the rate of resonant absorption taking place in the specimen as a function of energy, which is proportional to the relative velocity of the drive. Each channel which corresponds to a certain relative velocity is opened for an equal interval of time to the counter. The periodical scan through all channels is controlled by a wave form generator, which produces the velocity reference signal and two other digital pulses: the starting pulse (STA) and the channel advancing pulse (CHA) which are the basic clocks for the system.

The entire system consists of the Mössbauer source, the absorber, γ- and X-ray detector, the drive, which moves the source or the absorber, the wave form generator, an embedded computer based on X86 architecture. The embedded computer just consists of an Intel 486 DX/4 CPU, 16M EDO memory, a general PC motherboard with the power supply, an 8M AFM-2000 IDE Flash disk, an NE2000 net card, a self-made data acquisition card based on the ISA bus architecture. There is no need for other devices, such as: Monitor, Hard-Disk, keyboard, etc. The experimental result shows that the system works very stably and operation is convenient and simple. Once it powers on, it can be freely controlled and visited through the Internet [5]. Considering the system's security, only authorized users could be allowed to control the equipment.

3. System software

The system is based on the client–server model and modular structure. It is composed of three parts: a real-time collect data module (named Rtcollect.o), a Daemon process module (named Reserve) which is running in the background and providing the interface between the real-time module and user interface, and Client part. Its structure is shown in Fig. 1. The first two are running on Mini Real-Time Linux [4,8], a minimum sized Real-Time Linux that is small enough to boot from a single floppy (or small Flash memory device) into a ramdisk, all the codes are written in C, and the GNU gcc C compiler for Linux is employed. The client is written in the Java language to be platform-independent, and it does provide a convenient accessing interface for users.

(1) *Daemon process module*: This includes web server process, and the daemon process, which listens to the port, verifies the login of users, and monitors the status of the system. It also supports concurrent processes.

(2) *Real-time collect data module*: The main functions of real-time module (Rtcollect.o) are to process counts from the single-channel analyzer (SCA) by updating the appropriate channel, output the block data to the client part, and execute all kinds of commands from clients. In addition, it should record the system status, which will be reported back to the clients.

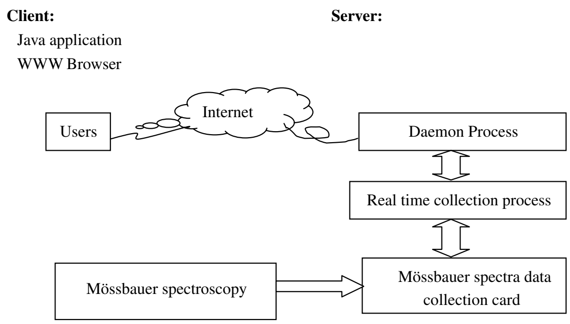

Fig. 1. Diagram of the system's software structure.

(3) *Client part*: There are two methods by which authorized users could control the system and access the data. One is by a WWW browser, and the other is through the Java application program (an example of a display is shown in Fig. 3).

4. Communication model

In this section the simple and effective communication model (Fig. 2) is presented briefly. It is the key part of this multi-task, multi-user, and real-time data acquisition system. There are three types of data being exchanged between the normal process and real-time module: the commands from the client, the result of the command being executed, and the collection of data. The first two are of the character data type. The last, the collecting data is of the block data type. So dealing with the first two just uses the pipes and real-time FIFO pipes. In this model, it is required to create an in-pipe and an out-pipe, and the same applies to real-time FIFO. One is used for receiving the commands; the other is used for returning the result. The collected data is exchanged between Reserve and Rtcollect.o via an mbuff shared memory module (mbuff.o) [9,10], which allows kernel tasks and user space processes to share large blocks of

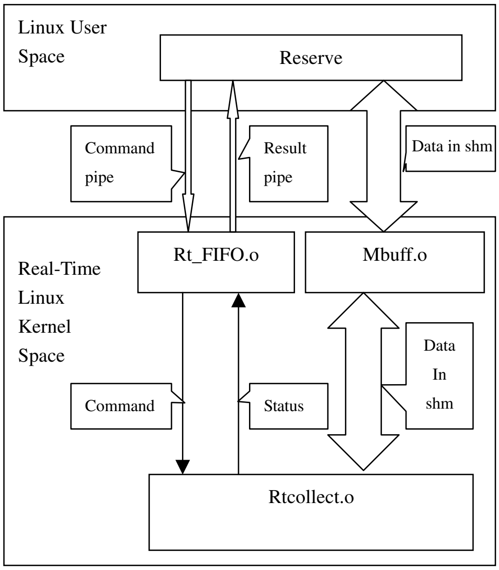

Fig. 2. The communication model.

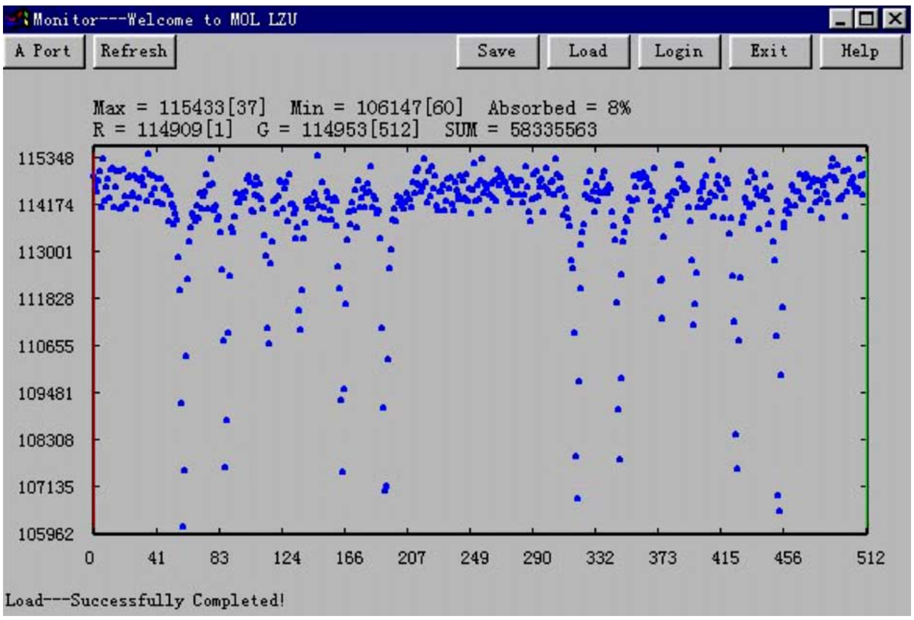

Fig. 3. The user application interface (the Mössbauer spectrum is of ^{57}Fe).

memory, in other words, even big blocks of data could be exchanged quickly between the kernel and user space.

5. Performance

The main parameters of the remote data acquisition system designed by us are

(1) channel range: 0–1024,
(2) maximum work frequency: 100 Hz,
(3) maximum drive frequency which determines the scanning time of each channel: 100 Hz (for 1024 channels),
(4) dwell time: $\leqslant 1.0$ μs, which is the interval between two counting pulses,
(5) maximum counting capacity: 3.6×10^9 (32 bits) (for 1024 channels),
(6) maximum counting frequency: 2 MHz,
(7) the system could be visited by ten clients at the same time through the Internet,
(8) collect two series of spectra simultaneously.

6. Experimental results

The remote data acquisition and control system has been employed at the Key Laboratory for Magnetism and Magnetic Material of Ministry of Education, Lanzhou University. A Mössbauer spectrum of ^{57}Fe, successfully collected by this embedded data acquisition system (the *x*-coordinate is the open channel, and the *y*-coordinate is the count corresponding to the channel), is presented in Fig. 3. Analysis of magnetic properties of the samples indicates that the system works well.

7. Conclusion

The design of the system described here is initiated after analysis of requirements for such a system in the PR China, and it is an example of an embedded data acquisition system, which satisfies the design criteria and is easy to operate. Also it enhances the capabilities of the technique of one Mössbauer spectroscopy's laboratory, making it as an intelligent machine which could be visited freely through Internet. The cost of this system as an addition to a previously existing Mössbauer spectroscopy is limited, just about \$150–2002 prices. Therefore it can be built in most laboratories.

Acknowledgements

I want to thank my supervisor Dr. Rongjie Zhou for his funding of this work, and thank for the generous help and encouragement from Nicholas McGuire, especially for the book –"Understanding the Linux Kernel", from which I have learned a lot and continue to learn.

References

[1] G.M. Kalvius, E. Kankeleit, Recent improvements in instrumentation and methods of Mössbauer spectroscopy, in: Proceedings of a panel on Mössbauer Spectroscopy and its Applications, STI/PUB/304, IAEA, Vienna, 1972.
[2] M.A. Plyer, F.W.D. Woodhams, J. Phys. E 11 (1978) 191.
[3] R. Zhou, J. Yang, F. Li, Nucl. Instr. and Meth. B 95 (1995) 249.
[4] N. McGuire, The second real-time Linux workshop, 2000.
[5] Q. Zhou, Department of Physics, Lanzhou University, Master Thesis, 2001.
[6] R. Zhou, F. Li, Z. Zhang, Hyperfine Interact. 24 (1988) P181.
[7] V. Yodaiken, RTLinux, real-time linux, FMSLab, Soccoro, New Mexico,USA, 1999.
[8] N. McGuire, Minirtl – a minimum realtime linux system, The first real-time Linux workshop, 1999.
[9] F.M. Proctor, Using Shared Memory in Real-Time Linux. Available from http://www.isd.cme.nist.gov/projects/emc/shmem.html.
[10] T. Motylewski, Sharing Memory Between Kernel and User Space in linux, 2000, http://crds.chemie.unibas.ch/linux/shm-orlando/.